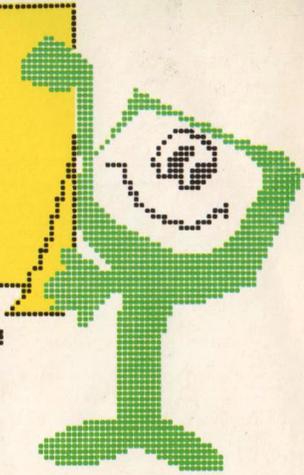
20 VIDEOLEZIONI DI BASIC
PER IMPARARE CON LO SPECTRUM

14

Spectrum

16K/48K/PLUS

VIDEO BASIC SPECTRUM
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore
Editoriale: Roberto Pancaldi
Autore: Softidea - Via Indipendenza 88 - Como
Redazione software:
Francesco Franceschini, Roberto Rossi,
Alberto Parodi, Luca Valnegri
Segretaria di Redazione:
Marta Menegardo
Progetto grafico:
Studio Nuovaidea - Via Longhi 16 - Milano
Impaginazione:
Silvana Corbelli
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Autorizzazione alla pubblicazione Tribunale di
Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70
(autorizzazione della Direzione Provinciale delle
PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo
Editoriale Jackson S.r.l. - Via Rosellini, 12
20124 Milano, mediante emissione di assegno
bancario o cartolina vaglia oppure
utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere
richiesti direttamente all'editore
inviando L. 10.000 cdu. mediante assegno
bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

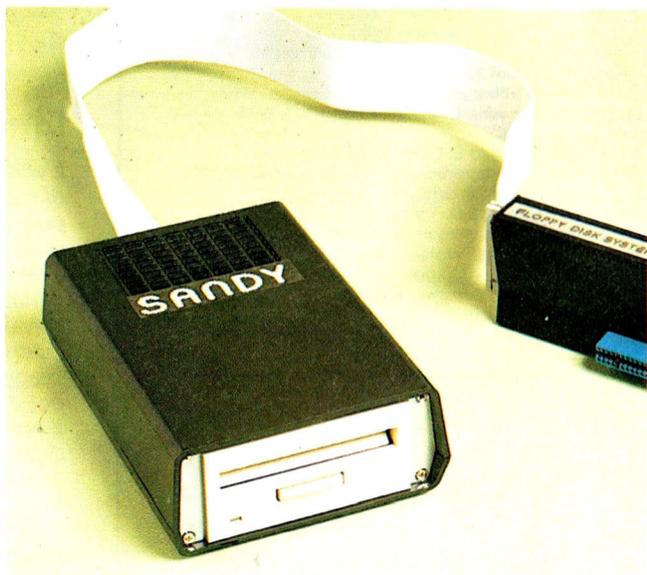
Gruppo Editoriale Jackson

SOMMARIO

Introduzione

L'economico registratore è la memoria di massa per eccellenza addirittura si può dire che l'home computer non esisterebbe senza registratore.
Che noia però i lunghi e interminabili minuti per il caricamento o salvataggio di un programma, che estenuanti attese nella ricerca di un programma e che scarsa affidabilità, poi, in tutte le operazioni che vedono il registratore coinvolto.
Come sempre la tecnologia ha dato una risposta a tutti questi problemi: i microdrive.
Trattare allora grosse moli di dati, in breve tempo e con una quasi totale affidabilità diventa semplice, il trucco è conoscere i comandi.

Il Microdrive

Abbiamo visto nelle scorse lezioni come sia possibile - attraverso il registratore a cassette - memorizzare su nastro magnetico tutti i programmi e i dati che andrebbero altrimenti irrimediabilmente perduti spegnendo il calcolatore. La registrazione e la lettura da nastro - sicuramente avrai già avuto modo di accorgertene - è un'operazione molto semplice, affidabile ed economica: tuttavia richiede lunghe attese per il caricamento delle informazioni e, soprattutto, necessita di un intervento manuale per avviare e spegnere i motori del registratore (il famoso "START TAPE"). La ricerca dei programmi sul nastro è inoltre affidata a tecniche abbastanza empiriche, come il conteggio dei secondi o il rilievo del numero di giri compiuti dalla rotella di trascinamento, con il continuo rischio (soprattutto quando si è abbastanza distratti) di sovrapporre pezzi già registrati e provocare la perdita di programmi che magari avevano richiesto lunghe ore di paziente battitura.

Una delle più interessanti possibilità per l'espansione delle prestazioni offerte dal tuo Spectrum è allora sicuramente costituita dall'unità a Microdrive. Lo scopo del Microdrive, del quale ci occuperemo nella nostra lezione, è infatti proprio quello di eliminare tutti questi "inconvenienti", permettendo una registrazione altrettanto sicura di quella offerta dal registratore a cassette, ma molto più veloce e, soprattutto, completamente automatica.

Al contrario di quando si usa il registratore, infatti, non è necessario preoccuparsi in alcun modo del funzionamento del drive: la gestione delle varie operazioni - come per esempio la registrazione o il caricamento dei programmi - avviene in questo caso sotto il diretto controllo dell'unità centrale, senza più alcun intervento esterno di ricerca o di verifica.

Come funziona un Microdrive

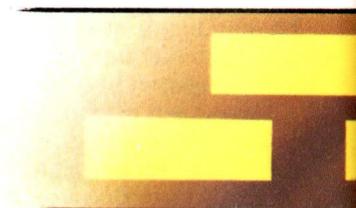
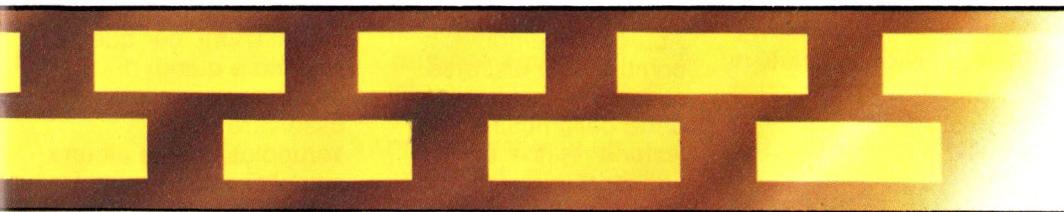
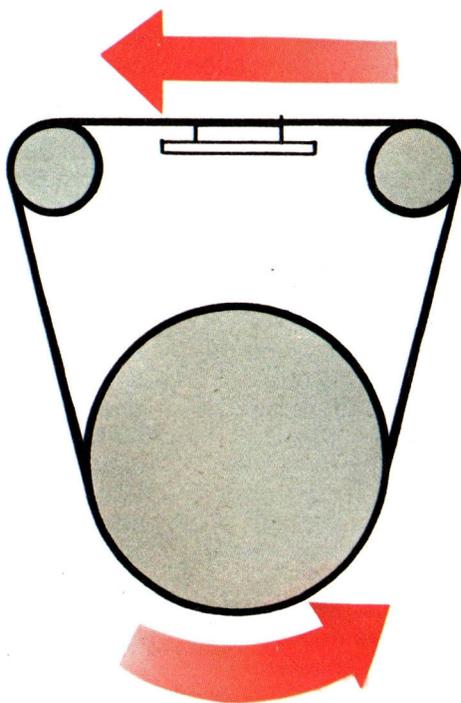
Un Microdrive consiste essenzialmente in un dispositivo, il cui funzionamento risulta molto simile a quello di un normale registratore a cassette, in grado di leggere o di scrivere - tramite un'apposita testina - il nastro magnetico contenuto nella cartuccia (o cartridge) che può essere inserita nel Microdrive stesso.
La cartuccia è costituita da un involucro plastico all'interno del quale si trova un nastro magnetico disposto secondo un ciclo continuo, cioè in maniera tale da non richiedere alcun avvolgimento per la lettura o la scrittura delle informazioni.
L'avanzamento del nastro può quindi avvenire sempre nella stessa direzione, consentendo così al sistema di raggiungere di velocità di lettura e di scrittura delle informazioni altrimenti irrealizzabili.
Tale soluzione comporta comunque anche qualche inconveniente, in particolar modo per quanto riguarda il tempo di accesso delle informazioni (intendendo con "tempo di accesso" quello necessario alla ricerca e al caricamento dei dati o dei programmi registrati).
Se infatti si desidera caricare un programma il cui inizio, per combinazione, è venuto

a trovarsi immediatamente dopo la testina, è necessario passare l'intero nastro (5 metri) prima che risulti possibile accedervi (dal momento che non è possibile riavvolgerlo), sprecando quindi tempo nella sola ricerca. Tieni comunque presente - per avere un'idea più chiara di cosa significhi "tempo" - che il nastro, grazie all'estrema velocità del motore, impiega circa sette secondi per compiere un ciclo completo. Tutta un'altra cosa rispetto alle comuni cassette audio!

È interessante soffermarci un attimo sulle modalità di registrazione dei dati sul nastro.

A differenza del normale registratore, infatti, con il Microdrive le informazioni vengono memorizzate secondo blocchi di dati di dimensioni fisse chiamati *settori*. Puoi quindi immaginare i settori come porzioni di nastro, in ciascuna delle quali è possibile scrivere dei dati.

La prima volta che il nastro viene adoperato, questi settori naturalmente non esistono: occorre allora eseguire una particolare operazione - chiamata *formattazione* - che permette appunto al Microdrive di costituire una sorta di sistema di riferimento, formato proprio da questi settori.

Al momento della formattazione, quindi, il nastro viene idealmente spezzettato in tanti segmenti, ciascuno dei quali pronto per registrare una ben determinata quantità di informazioni.

Cerchiamo adesso di capire quali possono essere il significato e l'utilità di una simile manovra.
Subito dopo la formattazione tutti i settori sono liberi; i successivi comandi di registrazione andranno allora via via a riempire uno dopo l'altro i vari settori, mentre i comandi di cancellazione li libereranno.
Il grosso vantaggio di aver formattato il nastro è costituito dal fatto che uno stesso programma non necessariamente dovrà essere registrato su settori adiacenti, permettendo così di sfruttare tutti i settori che

Disposizione dei dati sul nastro del microdrive.

L'anello di nastro contenuto in una cartuccia del microdrive è lungo 5 metri.

a seguito delle cancellazioni si sono venuti a liberare nei vari punti del nastro stesso. In altre parole, la formattazione consente di ottimizzare le risorse offerte dal nastro magnetico (eliminando cioè qualsiasi spreco di spazio e di tempo), permettendo inoltre ai circuiti di controllo del Microdrive di capire in qualsiasi istante in quale punto del nastro si trova la testina di lettura/scrittura. Approfondiremo comunque il discorso sulla formattazione nella parte della nostra lezione relativa al linguaggio.

Manutenzione delle cartucce

Le cassette del registratore dispongono di linguette, che se asportate, impediscono a chiunque cancellazioni o registrazioni accidentali. Anche le cartucce del Microdrive consentono questa opportunità: è sufficiente spezzare con un cacciavite o con qualcosa di simile la linguetta di plastica posta sul lato destro del cartridge, ed automaticamente sarà impedita la scrittura di qualsiasi nuova informazione sul nastro della cartuccia.
Se in un secondo tempo si desidera riutilizzarla è sufficiente chiudere lo spazio lasciato dalla linquetta con un pezzo di nastro adesivo.
Abbiamo visto che, nonostante le ridotte dimensioni, un minuscolo cartridge è in grado di registrare con precisione una grandissima quantità di informazioni; per questa ragione è quindi di importanza capitale osservare scrupolosamente alcune semplici, ma indispensabili precauzioni, analoghe a quelle già viste a proposito dei nastri magnetici, ma ancora più importanti, vista la relativa delicatezza delle cartucce:

- non toccare mai la superficie magnetica del nastro. Sulle dita è sempre presente un sottile strato di grasso, che può impedire il corretto contatto tra testina e nastro;
- evitare che la polvere si depositi sui cartridge: anche un microscopico granello di polvere o di cenere di sigaretta può infatti rovinare irrimediabilmente lo strato magnetico. La soluzione migliore è quella di riporre immdiatamente la cartuccia nel proprio astuccio protettivo non

appena terminato di usarla;
- non esporre le cartucce al sole, al caldo o alle basse temperature potrebbero verificarsi pericolose deformazioni del supporto plastico, con conseguenti malfunzionamenti nel trascinamento del nastri;
- allontanare il più possibile i cartridge da calamite, televisori, altoparlanti e in generale da qualunque apparecchiatura elettrica;
- tenere una copia di riserva di tutte le cartucce contenenti informazioni importanti, conservandole possibilmente in un luogo diverso da quello incui si trovano gli originali;
- quando il computer è spento non lasciare mai la cartuccia nel drive: estrarla sempre prima di spegnere la macchina e riporla immediatamente.

Prima di concludere il nostro discorso è bene ricordare che conviene sostituire immediatamente le cartucce che, in seguito a usura o cadute, dovessero presentare malfunzionamenti di un certo rilievo (come inceppamenti o bloccaggi nella rotazione del drive): il tuo Microdrive te ne sarà sicuramente grato, ripagandoti con un lavoro sempre veloce e, soprattutto, affidabile.

I canali

Lo Spectrum dispone di una struttura molto sofisticata per dialogare con le varie periferiche di ingresso ed uscita, basata principalmente sui flussi e sui canali. Tale struttura viene raramente utilizzata (salvo casi particolari) sullo Spectrum in configurazione standard (cioè priva di interfaccia 1), visto che sono pochi i dispositivi privi di comandi appositamente creati per il loro uso. Per esempio, l'invio delle informazioni allo schermo può essere fatto con l'istruzione PRINT, mentre per inviare qualcosa alla stampante si può utilizzare il comando LPRINT.
L'aggiunta del Microdrive richiede

tuttavia l'approfondimento di questo argomento, visto che la flessibilità offerta da questa periferica evidenzia in tutta la loro utilità i vantaggi di una simile struttura.

Un modo abbastanza semplice per capire il significato e la differenza tra flusso e canale è quello di considerare una qualsiasi unità periferica come suddivisa in due parti separate.
La prima, fisicamente visibile, è quella che invia o riceve le informazioni dall'unità centrale; si chiama canale. Esempi di canali verso i quali lo Spectrum può trasmettere sono lo schermo video o la stampante; i canali dai quali l'unità centrale può invece ricevere dati sono la tastiera o il registratore.
La seconda parte riguarda invece i percorsi che i dati devono seguire per giungere o partire dai canali. Tali percorsi si chiamano flussi. Sullo Spectrum esistono 16 distinti flussi disponibili, quattro dei quali vengono automaticamente riservati, al momento dell'accensione, a determinati canali.

NUMERO FLUSSO UTILIZZO INPUT	IDENTIFICAZIONE CANALE	UTILIZZO OUTPUT
# 0 tastiera	"K"	schermo infer.
# 1 tastiera	"K"	schermo infer.
# 2 ///	"S"	schermo super.
# 3 ///	"P"	stampante

I flussi 0 e 1 sono identici e sono connessi al canale "K". Tale canale invia in uscita le informazioni alla parte bassa dello schermo (cioè dove compaiono i messaggi di errore e di

INPUT) e riceve in ingresso i dati dalla tastiera. Il flusso 2 è invece in comunicazione con il canale "S", e stampa i caratteri sulla parte superiore dello schermo; il flusso 3 è connesso al canale "P", ed invia le informazioni alla stampante.
Non è possibile utilizzare i flussi 2 e 3 per l'input: qualsiasi tentativo di leggere qualcosa, per esempio dal flusso 3, produrrà il messaggio di errore

```
INVALID I/O DEVICE
```

Il simbolo del flusso è il cancelletto (#).
Per il momento faremo riferimento allo Spectrum non ancora connesso al Microdrive e all'interfaccia 1.

OPEN/CLOSE

Il concetto di flusso è abbastanza semplice da comprendere; può tuttavia darsi che tu ti stia domandando come fare e quale utilità possa avere associare dei numeri di flusso con dei dispositivi di ingresso/uscita. Avrai la risposta entro poco tempo. Innanzitutto diciamo che l'unità centrale, prima di poter dialogare con una periferica attraverso un flusso deve aprire (OPEN) questo flusso. L'apertura del flusso assolve quindi a un duplice scopo.
Innanzitutto associa un numero di flusso a un particolare dispositivo di I/O; in secondo luogo segnala il dispositivo di I/O che dovrà essere utilizzato.
Per aprire un flusso il BASIC dello Spectrum fornisce il comando

```
OPEN # F, C
```

dove "F" è il numero del flusso da aprire, mentre "C" è la stringa che specifica il canale da associare a tale flusso. Una volta impartito questo comando qualsiasi dato inviato o letto attraverso il flusso appena aperto si riferirà al canale "C".
Così

```
OPEN # 5, "K"
```

aprirà il flusso 5, associando ad esso la tastiera.
Adesso, se impartiamo questo comando:

```
INPUT # 5; A, B
```

potremo assegnare i valori alle variabili A e B nella stessa maniera di una normale istruzione INPUT, visto che il flusso è stato aperto verso la tastiera (KEYBOARD).

Se invece eseguiamo:

```
PRINT # 5; "CIAO A TUTTI"
```

invieremo la stringa "CIAO A TUTTI" al flusso 5, che corrisponde all'area inferiore dello schermo. Siamo così riusciti a scrivere con una istruzione PRINT nella parte dello schermo normalmente riservata ai messaggi di INPUT. Sfortunatamente, eseguendo il comando, la stringa non è visibile (la parte inferiore dello schermo viene infatti "pulita" al momento dell'arresto del programma).
Per vedere gli effetti di questo flusso prova allora ad eseguire questo breve programma:

```
10 OPEN # 5, "K"
20 PRINT # 5; RND
30 GOTO 20
```

Stavolta l'effetto risulterà ben visibile! Il programma tuttavia si arresterà con un messaggio di errore, visto che l'area di input non esegue lo scrolling nella stessa maniera della normale area di stampa.
In linea di principio ciascun flusso può funzionare sia in ingresso che in uscita; in pratica, il solo canale in grado di accettare dati sia in ingresso che in uscita è quello che abbiamo appena visto nell'esempio, cioè il canale della tastiera. Gli altri due - schermo e stampante - sono canali di solo output, e qualsiasi tentativo di leggere qualcosa da essi provocherà un messaggio di errore.
Questa è una limitazione dovuta interamente all'hardware connesso al flusso; in altre parole è indipendente dal flusso.
Per disconnettere un flusso ad un canale esiste un'altra istruzione, CLOSE, il cui scopo è proprio quello di separare ciò che OPEN aveva unito.
Nel nostro precedente esempio avremmo dovuto quindi scrivere:

```
CLOSE # 5
```

Avevamo infatti aperto il flusso numero 5. Adesso lo abbiamo chiuso.
CLOSE effettua una operazione che apparentemente sembra inutile, ma che invece risulta importantissima per il buon funzionamento del

computer. La chiusura dei flussi consente infatti di liberare dall'attesa il dispositivo connesso a quel flusso, evitando possibili errori o malfunzionamenti.

Come si usano i flussi

L'unico vantaggio che abbiamo sinora potuto apprezzare nell'uso dei flussi è stato quello di riuscire a scrivere nella parte inferiore dello schermo. Esiste tuttavia una serie di ottime ragioni per usare i flussi, la prima e più importante delle quali si chiama "indipendenza dalla periferica". Questo termine, apparentemente senza senso, è facilmente illustrabile con un esempio.
Supponiamo di voler scrivere un programma che produca una serie di numeri casuali sullo schermo o, in alternativa, sulla stampante.
Utilizzando PRINT e LPRINT il programma potrebbe essere una cosa del genere:

```
10 INPUT "PRINTER O SCHERMO"; A$
20 IF A$ (1) = "P" THEN LPRINT RND
30 IF A$ (1) = "S" THEN PRINT RND
40 GOTO 20
```

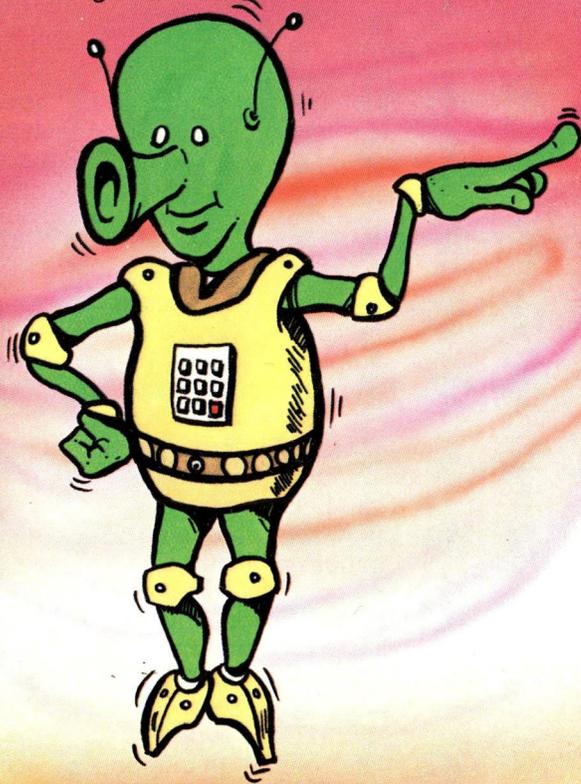
Con il flusso potremmo invece scrivere

```
10 INPUT "PRINTER O SCHERMO"; A$
20 OPEN # 5, A$ (1)
30 PRINT # 5; RND
40 GOTO 30
```

Per quanto questo esempio risulti troppo breve per essere realmente convincente, puoi essere sicuro che in lunghi programmi l'uso dei flussi è estremamente pratico e vantaggioso.
Usando il Microdrive i flussi diventano comunque inevitabili: speriamo che questa "introduzione" ai flussi ti abbia ben disposto al loro uso!

I file

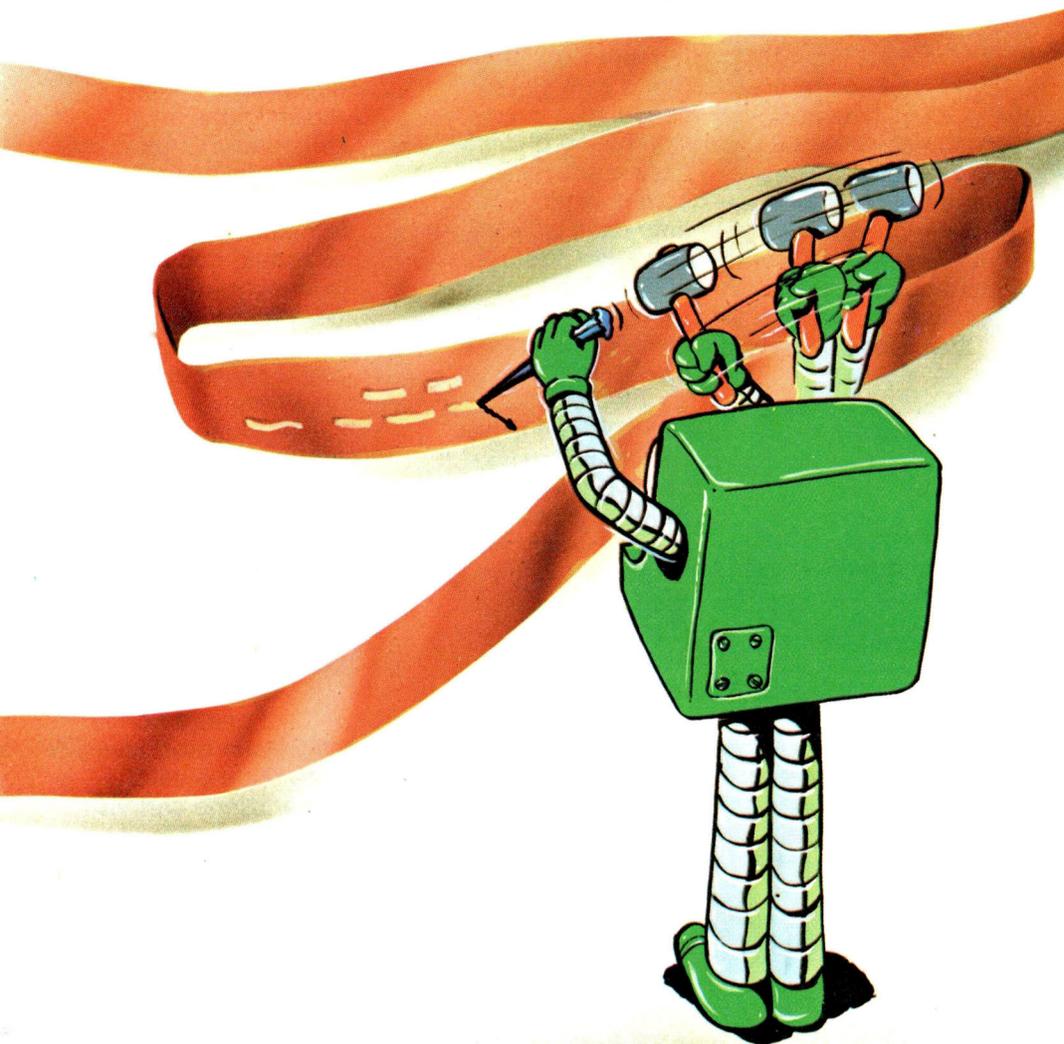
I programmi sono abbastanza utili, ma, diciamo la verità, quello che realmente si vuole dal calcolatore è trattare informazioni, dati, numeri, nomi, indirizzi, quantità.
Qualunque istruzione che possa essere scritta può anche essere inserita nella memoria del computer e, di conseguenza, memorizzata su cartridge.
Abbiamo tutti una qualche esperienza di archiviazione, sia che si tratti di archivi organizzati in un ufficio o di ricevute dei versamenti di conto corrente della banca o

ancora di ricette dei piatti preferiti. Sappiamo perfettamente quanto sia importante archiviare con ordine le nostre cose.
Un file è un insieme di informazioni che, per un motivo qualunque, si desidera raggruppare.
Esempi di file possono essere: i nomi degli alunni di una classe, i numeri estratti durante una partita di tombola, i nomi e gli indirizzi dei nati in un certo giorno, un programma.
Il Microdrive permette sul tuo Spectrum la creazione di file di tipo cosiddetto sequenziale.
Tale denominazione deriva dal fatto che tutti gli elementi che compongono il file vengono registrati uno di seguito all'altro, cioè in ordine sequenziale.
La ragione di questa soluzione è facilmente spiegabile con la costituzione fisica del supporto utilizzato dal drive: il lungo nastro magnetico all'interno del cartridge costituisce

infatti la struttura ideale per contenere informazioni secondo un simile ordine.
Esamineremo quindi come dovremo fare per memorizzare su cartuccia - raccogliendoli tra loro - tutti i dati e le informazioni che desideriamo non siano irrimediabilmente perduti spegnendo il calcolatore.
Per fare questo è comunque inizialmente necessario introdurre due nuovi termini, usatissimi quando si parla di file: Record e Campi.
Facciamolo (per essere più chiari possibile) attraverso un esempio.
L'elenco telefonico della tua città, preso nel suo insieme, è un file: esso contiene infatti informazioni ordinate e classificate secondo un ordine ben preciso.
Questo file è costituito da migliaia di record, ciascuno dei quali formato da un cognome, un nome, un indirizzo e un numero di telefono.
Un record può, essere quindi definito come un gruppo di dati correlati fra loro; le singole registrazioni che si susseguono nel record si chiamano invece campi.
Nel nostro caso esistono 4 campi per ciascun record: un campo Cognome, un campo Nome, un campo Indirizzo e un campo Numero di telefono.
La scelta del formato del record (cioè del numero di campi) è naturalmente arbitraria: avremmo potuto considerare - oltre ai campi appena visti - anche un campo Numero di prefisso, indicante appunto il prefisso telefonico.
Abbiamo detto che nei file sequenziali (come dice il nome stesso) le informazioni sono registrate una di seguito all'altra. Ciò significa che per leggere (o scrivere) l'ultima informazione del file occorre aver prima letto (o scritto) tutte le informazioni precedenti.
Questa limitazione può sembrare un notevole svantaggio (sarebbe alquanto scomodo se anche noi per cercare "Rossi Mario" nell'elenco telefonico dovessimo cominciare dalla prima pagina), tuttavia - grazie alla notevole velocità di lettura dei dati sul drive - entro file di dimensioni abbastanza contenute il tempo di ricerca di un singolo record è abbastanza ragionevole.
Verificheremo e approfondiremo comunque il nostro discorso nella parte della lezione dedicata alla programmazione.

FORMAT

La prima cosa da fare per usare una cartuccia vergine (cioè mai usata) è eseguirne la formattazione.

Qualsiasi tentativo di utilizzarla prima di aver eseguito questa operazione provocherà infatti l'insorgere del messaggio di errore:

```
MICRODRIVE NOT PRESENT
```

In altre parole, il Microdrive non riconoscerà la cartuccia. Per poterla formattare occorre allora innanzitutto inserirla nel Microdrive e quindi scrivere:

```
FORMAT "m"; 1; "nome"
```

Appena schiaccerai il tasto ENTER sentirai il motore avviarsi e per circa 30 secondi vedrai lo schermo prima lampeggiare, poi rimanere stabile, poi lampeggiare di nuovo. Al termine comparirà il messaggio OK: l'operazione di formattazione a questo punto sarà stata eseguita.
Cerchiamo di capire cosa è successo, cominciando innanzi tutto dalla sintassi del comando. L'istruzione FORMAT è - ovviamente - il comando di formattazione del nastro, mentre la "m" (nota le virgolette) specifica che l'istruzione è rivolta verso il Microdrive. Il numero 1 indica che il nastro si trova nel Microdrive 1; con altri Microdrive potrà assumere un altro valore (compreso tra 1 e 8).
Il nome, anch'esso racchiuso entro virgolette, è quello che si desidera dare alla cartuccia, cioè è una specie di titolo che da quel momento in avanti la contraddistinguerà dovrà essere una stringa con lunghezza massima di 10 caratteri.
Durante i 30 secondi il calcolatore "configurerà" la cartuccia, memorizzandovi il nome che avrai specificato con la FORMAT e controllando più volte ciascuna sezione del nastro magnetico.Se qualche sezione risulta danneggiata, essa viene segnata come inutilizzabile, in modo che le successive operazioni di registrazione la eviteranno. Controllare con molta attenzione, prima della formattazione, di avere sistemato con precisione la cartuccia all'interno del Microdrive: accade infatti abbastanza spesso che a causa di un incorretto posizionamento della testina alcuni spezzoni di nastro vengano considerati (erroneamente) danneggiati.
La cosa più importante da fare prima della formattazione è tuttavia quella di essere sicuri che sul nastro non si trovi alcun programma importante o utile. L'operazione cancella

infatti qualsiasi informazione contenuta in precedenza nella cartuccia, eliminandola in maniera definitiva. Conviene fare sempre molta attenzione prima di formattare un nastro, visto che molte volte si ricorre a questo comando per rendere il nastro "come nuovo".

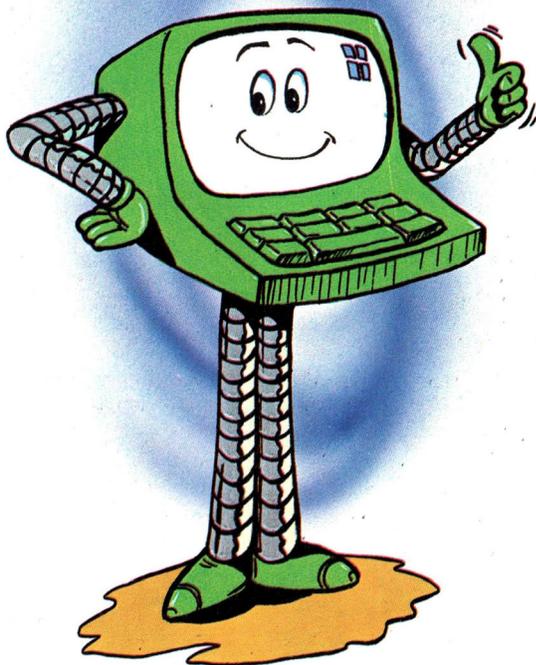
Sintassi del comando

FORMAT "m"; numero drive; "nome"

CAT

Dopo aver formattato la cartuccia potrai esaminarne il contenuto, impartendo il comando:

```
CAT 1
```

(l'1 dopo il CAT indica, come prima, il Microdrive 1).
Dopo qualche secondo (circa sette) vedrai infatti apparire sullo schermo video il nome che in precedenza avevi assegnato alla cartuccia, seguito da un numero. Questo indica quanti

LINGUAGGIO

kilobyte hai a tua disposizione sulla cartuccia per le successive operazioni di memorizzazione.
Secondo il manuale del Microdrive la capacità di una cartuccia è normalmente compresa tra gli 85 ed i 100 Kbyte; di solito essa si aggira comunque intorno ai 90 K.
Il comando CAT consente quindi di vedere ciò che è memorizzato sul nastro: per il momento, non avendo ancora registrato nulla, indicherà solamente il nome della cartuccia e lo spazio disponibile. Tra poco, quando avrai imparato come memorizzare i programmi, potrai comunque vederlo completamente in funzione.

Sintassi del comando

CAT numero drive

SAVE

Il grosso vantaggio nell'utilizzo del Microdrive si rivela in tutta la sua dimensione attraverso questo comando. La differenza di velocità e di risposta tra il registratore a cassette e il Microdrive, come tra poco avrai modo di verificare, è infatti davvero notevole.
SAVE esegue le operazioni di scrittura sul Microdrive in modo praticamente identico a quello che hai sinora usato con il registratore. Esso richiede comunque un pochino più di attenzione in fase di

scrittura del comando. Mentre con il registratore era infatti sufficiente battere per esempio:

```
SAVE "PIPPO"
```

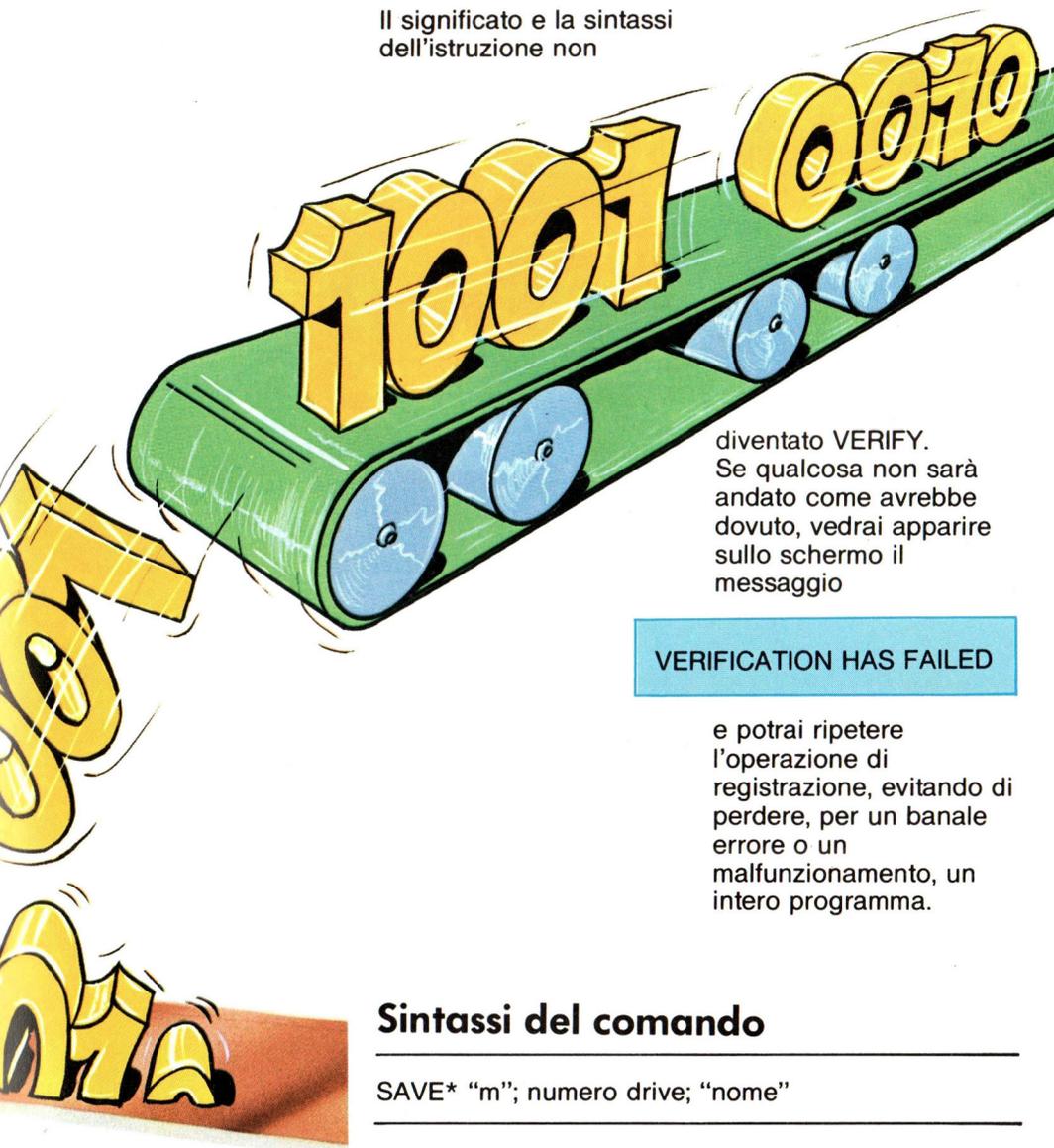
e il programma in memoria veniva salvato sulla cassetta con il nome PIPPO, ora bisogna specificare che si desidera memorizzare il programma sul Microdrive, indicando - analogamente a quanto visto prima - anche il numero del Microdrive. Il corrispondente comando per la scrittura sulla cartuccia dovrà quindi essere

```
SAVE*"m"; 1; "PIPPO"
```

Da notare l'asterisco subito dopo il SAVE. La "m" denota che l'operazione deve essere fatta con il Microdrive; l'1 individua invece quale dovrà essere il drive.
La piccola fatica aggiuntiva richiesta per battere l'istruzione - leggermente più lunga di quella relativa al registratore - viene comunque ampiamente ripagata al momento dell'ENTER.
La procedura di registrazione è infatti completamente automatica (niente più START TAPE, THEN PRESS AND KEY) e, soprattutto, velocissima. Nel breve periodo in cui sentirai girare il motore del drive (eventuali rumori anomali indicano il più delle volte un errato posizionamento della cartuccia), e vedrai accendersi la luce rossa, il tuo programma verrà memorizzato sul supporto magnetico. Un successivo comando CAT ti potrà convincere dell'avvenuta registrazione.
Per accertare che tutto sia andato per il verso giusto è comunque sempre presente il comando VERIFY. Nel

nostro esempio si dovrà scrivere

VERIFY* “m”; 1; “PIPPO”

Il significato e la sintassi dell’istruzione non dovrebbero esserti più di difficile comprensione: è tutto uguale a prima, eccetto il SAVE, che è diventato VERIFY.

Se qualcosa non sarà andato come avrebbe dovuto, vedrai apparire sullo schermo il messaggio

VERIFICATION HAS FAILED

e potrai ripetere l’operazione di registrazione, evitando di perdere, per un banale errore o un malfunzionamento, un intero programma.

Sintassi del comando

SAVE* “m”; numero drive; “nome”

LOAD

Una volta salvato il programma su cartuccia è naturale che si voglia avere anche la possibilità di riutilizzarlo. Per questo esiste LOAD. Parlando sempre del nostro programma PIPPO, l'istruzione di lettura dal Microdrive dovrà allora essere:

```
LOAD* "m"; 1; "PIPPO"
```

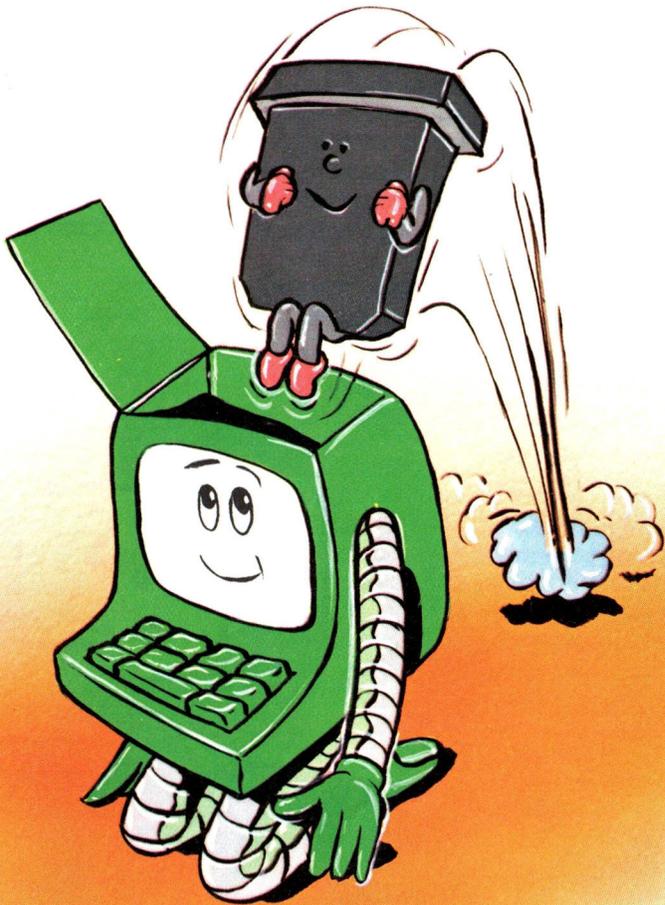
Anche qui - a parte eventuali errori di battitura o di posizionamento della cartuccia - dopo pochi secondi di funzionamento tutto sarà stato fatto: il programma sarà cioè caricato nella memoria del tuo Spectrum.
A questo punto non abbiamo più bisogno di dilungarci oltre: hai già capito che tutto ciò che si può eseguire sul registratore, con le dovute modifiche, è possibile farlo anche sul Microdrive. L'unica eccezione è costituita dal fatto che non si può più tralasciare di scrivere il nome del programma, come invece era consentito con il registratore. Così:

```
LOAD* "m"; 1; " "
```

non caricherà più il primo programma presente sulla cartuccia (anche perché, non esistendo più un principio o una fine del nastro, non esisteranno nemmeno il primo e l'ultimo programma), ma provocherà l'insorgere del messaggio di errore

```
INVALIDE NAME
```

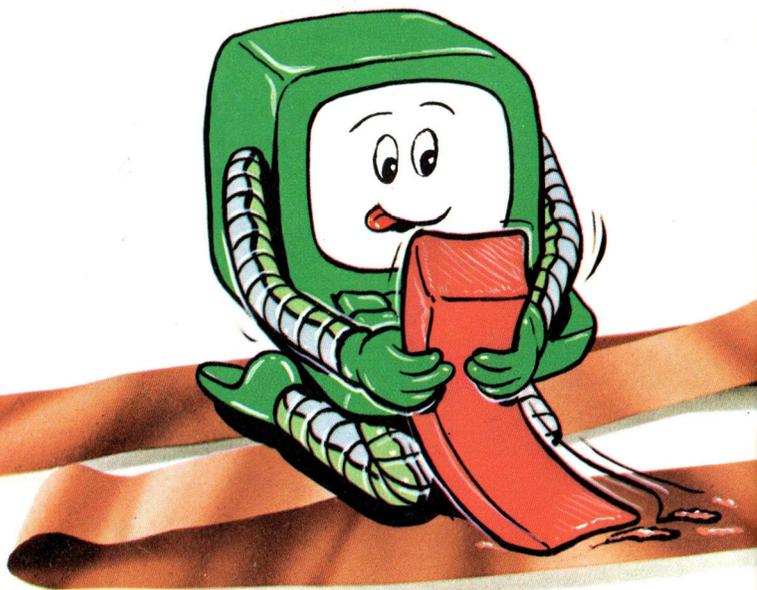
Sintassi del comando

LOAD* "m"; numero drive; "nome"

ERASE

Continuando il paragone tra registratore e Microdrive, vediamo innanzi tutto a cosa serve ERASE.
Supponiamo di avere memorizzato sia su cassetta che su cartuccia uno stesso programma, per esempio - tanto per cambiare - il nostro PIPPO. Ad un certo punto ci accorgiamo che non ne abbiamo più bisogno. Cancellarlo dalla cassetta è un attimo: basta riavvolgere il nastro e registrarci sopra qualcosa d'altro. Ma con il Microdrive?

Proprio per questo esiste ERASE! Questo comando elimina infatti dal cartridge il programma o, più in generale, il file che avremo specificato. Vediamolo con un esempio. Eseguendo

```
ERASE "m"; 1; "PIPPO"
```

cancelleremo il file PIPPO dal nastro della cartuccia posta nel Microdrive 1.
Come puoi notare, a

differenza degli altri comandi del Microdrive, ERASE non richiede l'asterisco. Naturalmente, perché sia possibile cancellarlo, il file deve essere presente sulla cartuccia; tentativi di cancellare file inesistenti porteranno inevitabilmente al messaggio:

```
FILE NOT FOUND
```

(file inesistente)

L'operazione di cancellazione va sempre fatta prima di aggiornare un qualsiasi programma. Se infatti cercherai di memorizzare una nuova versione di PIPPO senza aver precedentemente cancellato quella vecchia, otterrai un errore del tipo:

```
WRITING TO A "READ" FILE
```

(stai cercando di scrivere su un file di "LETTURA")

Così, tutte le volte che vorrai registrare un programma il cui nome è già presente nella cartuccia avrai due possibilità: o cambiare il nome del file da registrare, oppure cancellare preventivamente il vecchio file dal Microdrive.

Sintassi del comando

ERASE "m"; numero drive; "nome"

Usare i file sequenziali

In questa parte della lezione approfondiremo il discorso che poco fa avevamo cominciato a proposito dei file. Innanzitutto, prima di poter accedere a un file bisogna aprirlo.
Aprire un file significa comandare al nostro Spectrum di ricercare delle informazioni relative a questo file: se esso è sul disco e, se lo è dove si trova esattamente.
Il comando OPEN fa eseguire al drive tutte queste operazioni, riservando anche una certa area nella memoria del computer che verrà usata come buffer (detta anche memoria tampone). Il buffer non è altro che un serbatoio di informazioni il cui compito è di permettere al drive di accedere solo saltuariamente alla cartuccia e non ogni volta che il programma ha un singolo dato da leggere o da scrivere. In altre parole, l'accesso al nastro avviene per blocchi di informazioni e non per singolo dato, con conseguente grande risparmio di tempo. Un esempio di comando OPEN per l'apertura di un file sequenziale è il seguente:

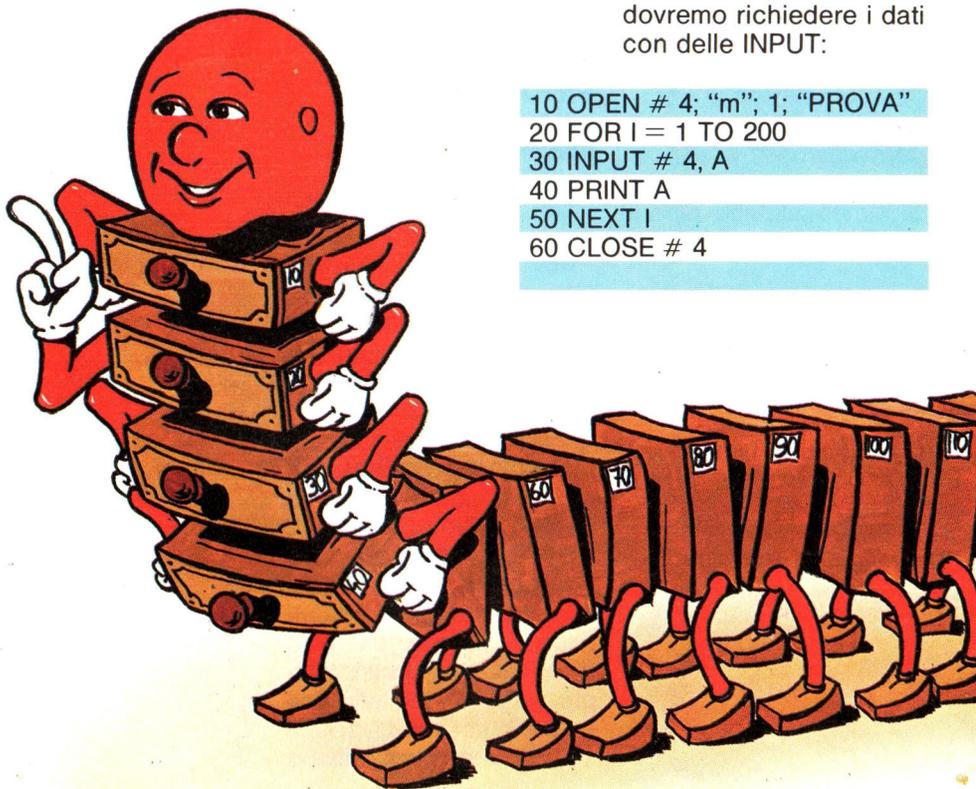
```
OPEN # 4; "m"; 1; "PROVA"
```

A questo punto tutte le istruzioni PRINT # che si riferiranno al canale appena aperto provocheranno la memorizzazione di dati sul cartridge. Questo breve programma illustra il funzionamento di quanto abbiamo appena detto:

```
10 OPEN # 4; "m"; 1;
     "PROVA"
20 FOR I = 1 TO 200
30 LET A = RND
40 PRINT A
50 PRINT # 4; A
60 NEXT I
70 CLOSE # 4
```

Dopo l'apertura del file verranno inviati 200 numeri casuali rispettivamente al video e al drive; alla fine del ciclo il file sarà chiuso e i numeri saranno stati memorizzati sul nastro magnetico.
Subito dopo l'ultimo numero, in seguito alla CLOSE, verrà inoltre memorizzato un carattere di EOF (End OF File, cioè fine del file)

necessario al computer per riuscire a gestire in futuro quel file, quando vorrai recuperare ciò che vi era scritto.
L'importanza di CLOSE, oltre che per la scrittura del carattere EOF, è rilevabile anche osservando questo programma al lavoro.
Potrai infatti renderti conto che la spia rossa del drive non rimarrà accesa in continuazione: le informazioni verranno infatti registrate solo quando il buffer sarà stato man mano riempito nel corso dell'esecuzione. Se non avessimo inviato la CLOSE, i dati rimasti, contenuti nel buffer al termine del ciclo, non avrebbero mai raggiunto la cartuccia, visto che il computer si sarebbe aspettato nuovi "arrivi", ed il file sarebbe risultato incompleto.
Quindi occorre sempre ricordarsi di chiudere i file.
Vediamo adesso come fare per recuperare i numeri che abbiamo appena memorizzato sul disco. Anziché inviare delle PRINT, ora dovremo richiedere i dati con delle INPUT:

```
10 OPEN # 4; "m"; 1; "PROVA"
20 FOR I = 1 TO 200
30 INPUT # 4, A
40 PRINT A
50 NEXT I
60 CLOSE # 4
```

Il funzionamento del programma sarà identico a quello visto prima, tranne che il computer - anziché scrivere - leggerà dal disco.
Un'importanza particolare rivestono, al momento della scrittura dei file, i caratteri separatori tra i vari campi. Per poter leggere i dati con l'istruzione INPUT è infatti necessario che i campi siano registrati, facendo comparire dopo ciascuno di essi un carattere separatore valido, cioè riconosciuto come tale dal drive.
Il principale carattere separatore riconosciuto è il CARRIAGE RETURN (CHR$ (13)). Esso può essere inserito con la funzione CHR$ (13), oppure facendo terminare la istruzione PRINT senza punteggiatura (come abbiamo fatto noi nell'esempio appena visto), oppure separando i vari elementi dalla PRINT con un apostrofo (che è l'equivalente di un ENTER).
La cosa migliore, senza entrare troppo in inutili dettagli, è comunque quella di non eseguire mai PRINT od INPUT multiple (cioè del tipo PRINT # 4, A, B, C o INPUT # 4, A, B, C), ma ricorrere sempre a più istruzioni semplici. Per quanto i programmi diventino leggermente più lunghi, la facilità di uso di una simile soluzione permette di restare sempre padroni della situazione, almeno sino a quando non si è acquistata sufficiente familiarità con i segreti dei file.
Un'altra cosa da segnalare è che l'aggiornamento di un file sequenziale può essere ottenuto soltanto riscrivendo completamente il file. Infatti la presenza del carattere EOF, inserito subito dopo l'ultimo record, impedisce l'inserimento di nuovi elementi in fondo alla lista.

Le uniche cose da fare sono allora:
— aprire in lettura il file da aggiornare;
— aprire in scrittura un nuovo file;
— copiare ordinatamente i record dal vecchio file nel nuovo apportando via via le modifiche, le cancellazioni o le aggiunte;
— chiudere alla fine i due file, cancellando il vecchio.

Vediamo quindi come avremmo dovuto fare per aggiungere altri 100 numeri casuali al file che avevamo creato prima:

```
 10 OPEN # 4; "m"; 1; "PROVA"
 20 OPEN # 5; "m"; 1; "PROVA2"
 30 FOR I = 1 TO 200
 40 INPUT # 4; A
 50 PRINT # 5; A
 60 NEXT I
 70 CLOSE # 4
 80 ERASE "m"; 1; "PROVA"
 90 FOR I = 1 TO 100
100 PRINT # 5; RND
110 NEXT I
120 CLOSE # 5
```

L'esempio che abbiamo appena fatto era abbastanza semplice, visto che sapevamo perfettamente quanti numeri dovevamo trasferire dal vecchio al nuovo file prima di passare alla estrazione di nuovi numeri casuali. Il più delle volte il numero di letture che possono essere effettuate prima di arrivare all'EOF è invece sconosciuto: tieni inoltre presente che cercare di leggere oltre l'ultimo record provoca un messaggio di errore. Non è quindi possibile leggere i record "a casaccio".

Un metodo quasi infallibile è allora il seguente. Bisogna modificare il programma, in modo da scrivere nel file, come ultimo record, uno o più caratteri speciali, o facilmente riconoscibili, che tu stabilirai essere il segno di fine-file (per esempio, con il nostro file di numeri avremmo potuto scrivere un numero negativo; con un file di stringhe potremmo invece scrivere una stringa di segnalazione con un significato inequivocabile, come "Z * Z * Z").

In questo modo in fase di lettura basterà inserire una istruzione di controllo di fine file, che risulterà verificata al momento in cui il carattere speciale verrà incontrato dal programma.

Ecco un esempio in proposito. Il programma che segue scriverà sulla cartuccia un file, composto da tante stringhe quante ne vorrai introdurre dalla tastiera. Quando deciderai di terminare basterà che tu batta solo ENTER. Prima della chiusura del file il computer scriverà automaticamente come ultimo record il nostro carattere speciale "Z * Z * Z".

```
10 OPEN # 4; "m"; 1; "PROVA3"
20 INPUT "SCRIVI LA STRINGA (SOLO ENTER PER FINIRE)"; A$
30 IF A$ = " " THEN GOTO 60
40 PRINT # 4; A$
50 GOTO 20
60 PRINT # 4; "Z * Z * Z"
70 CLOSE # 4
```

Potrai adesso recuperare e rileggere le stringhe che hai appena battuto, con questo secondo programma:

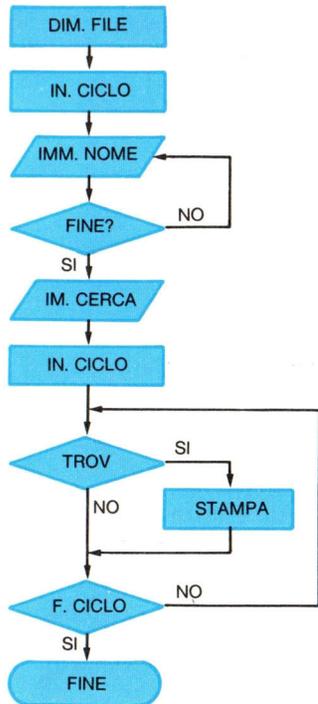
```
10 OPEN # 4; "m"; 1; "PROVA3"
20 INPUT # 4; A$
30 IF A$ = "Z * Z * Z" THEN CLOSE # 4 : STOP
40 PRINT A$
50 GOTO 20
```

Adesso la conoscenza del numero di elementi appartenenti al file non è più necessaria: il nostro "Z * Z * Z" avverte infatti l'unità centrale di non tentare di leggere oltre, perché non esistono altri

record. In questo modo ci siamo messi in salvo da un'eventuale possibilità di errore.

Ricerca su file

Il programma analizzato questa volta riguarda un'applicazione tipicamente informatica: la memorizzazione e la ricerca di informazioni. Nella fase preliminare memorizzeremo 9 informazioni qualsiasi (nomi, parole, numeri). Successivamente rintracceremo quelle desiderate in base ad un argomento anche parziale di ricerca.

```
10 DIM F$ (9,10)
20 FOR I=1 TO 9
30 INPUT "NOME"; F$ (I)
40 NEXT I
50 INPUT "CERCA"; C$
60 FOR I=1 TO 9
70 IF F$ (I, TO LEN C$) = C$ THEN PRINT F$ (I)
80 NEXT I
```

L'errore è sempre in attesa: malignamente aspetta una tua distrazione, anche minima. Non dargli mai la possibilità di sogghignare. Impara la sintassi delle istruzioni. Realizza programmi in forma modulare, facilmente leggibili, e molto documentati. Nella digitazione, poi, metti la massima attenzione: anche una virgola al posto sbagliato, compromette la buona riuscita del programma.

VIDEOESERCIZI

Dopo aver esaminato attentamente il listato rispondi alle domande verificando le tue risposte personalmente alla tastiera.

```
 5 CLEAR 24500
10 GO SUB 50
20 GO SUB 10
30 RETURN
40 STOP
50 LET S = 32400
60 PRINT "QUANTE VARIABILI..."
70 GO TO 30
80 IF S < 56 THEN RETURN
90 RETURN
```

Quale errore causa questo programma?

La "IF" in linea 80 potrà mai essere vera?

Il programma si ferma con "STOP" ... o no?

Vista la CLEAR in linea 5 quanti GOSUB senza un RETURN potrà sopportare il GOSUB STACK?